PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/60

A1

(11) International Publication Number: WO 98/22897

(43) International Publication Date: 28 May 1998 (28.05.98)

(21) International Application Number: PCT/GB97/03118

(22) International Filing Date: 12 November 1997 (12.11.97)

(30) Priority Data:
96308478.5 22 November 1996 (22.11.96) EP
(34) Countries for which the regional or international application was filed: GB et al.

(71) Applicant *(for all designated States except US)*: BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

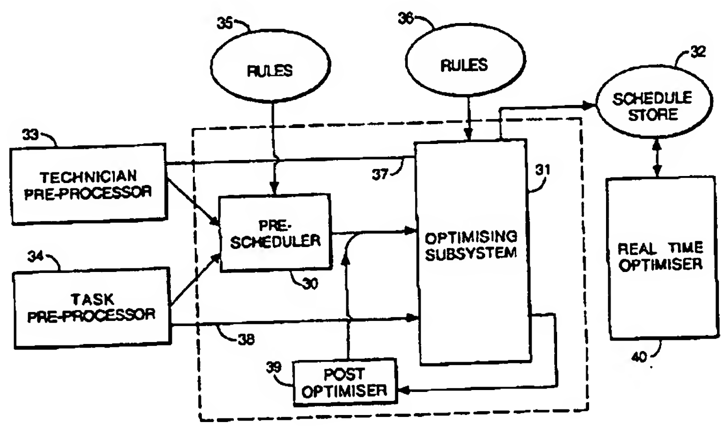
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: LESAINT, David [FR/GB]; 19 Surbiton Road, Ipswich, Suffolk IP1 4AX (GB). VOUDOURIS, Christos [GR/GB]; 5 Sandpit Close, Ipswich, Suffolk IP4 5UP (GB). AZARMI, Nader [GB/GB]; 1 Sonell Court, Wivenhoe, Colchester, Essex CO7 9RT (GB). LAITHWAITE, Robert, Noel, William [GB/GB]; 14 St. Mary's Park, Bucklesham, Near Ipswich, Suffolk IP10 0DY (GB). O'DONOGHUE, John, Joseph [IE/GB]; 18 Murray Road, Ipswich, Suffolk IP3 9AH (GB). NOBLE, Andrew, Paul [GB/GB]; 10 Burlington Avenue, Kew, Richmond, Surrey TW9 3DQ (GB). WALKER, Paul [GB/GB]; 213 Breightmet Drive, Bolton BL2 6NP (GB). ALLETSON, Ian, Stuart [GB/GB]; 12 Footward Crescent, Rochdale, Lancashire OL12 6TB (GB).

(74) Agent: LIDBETTER, Timothy, Guy, Edwin; BT Group Legal Services, Intellectual Property Dept., Holborn Centre, 8th floor, 120 Holborn, London EC1N 2TE (GB).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: RESOURCE ALLOCATION



(57) Abstract

A plurality of resources, typically service operatives, are allocated to a plurality of tasks by a method in which initial information relating to the tasks to be allocated and the resources available to perform the tasks is provided to first generate an initial series of schedules allocating resources to the tasks, and then modifying the individual schedule of at least one resource in response to updated information, whereby changes to individual schedules may be made in response to such updated information independently of the schedule generation means. The initial series of schedules may be generated in a two-stage process in which a rule-based system (30) allocates tasks selected as being difficult to allocate (e.g. because they are linked to other tasks), and then a stochastic (non-systematic) search system (31) compiles the rest of the schedule. Periodically, the stochastic system may be interrupted to allow a further rule-based system (39) to analyse the schedules created thus far, and fix the best ones in the schedule, so that the stochastic system (31) can then concentrate on improving the remaining schedules. In order to allow the system to handle rapid changes in the requirements for tasks and the resources, on a scale faster than the time required to generate the schedules, a schedule modification system (40) is arranged to make changes in the short term in between schedule updates delivered by the schedule generation system (30, 31).

RESOURCE ALLOCATION

This invention relates to a method for optimising the allocation of a plurality of resources to a plurality of tasks, and to an apparatus for performing such a method. It is particularly suited for use in situations where the availability of resources, and the tasks to be performed, both change dynamically. An example of such a situation is the allocation of tasks to a field force of personnel, for example ambulance or taxi drivers, a vehicle repair call-out field force, or a maintenance field force for a distributed system such as an electricity or water supply system or a telecommunications network.

In such situations the workload is highly variable and volatile, and tasks have to be allocated in real-time since the necessary response times are of the order of the duration of the tasks themselves, and very much shorter than a technician's working day. The durations of the individual tasks are themselves very variable and often unpredictable, which affects resource availability for those tasks awaiting allocation.

A prior art system, described in International patent application no. WO 95/26535, describes a system in which, for each resource, the time at which it will become available is estimated. Each task is assigned a time-dependent function, referred to hereinafter as a "cost function". This cost function is a measure of the consequences of a resource being allocated to the task at a given time. For example, if a resource fails to be allocated within a deadline which has been guaranteed to a customer, compensation may be payable to the customer. Travelling time to, from, and between tasks, and idle time (incurred if a resource cannot perform the next allocated task immediately the resource becomes available to perform it, for example before access to premises can be gained, or before a preceding task in a sequence has been done) are other factors. For each combination of tasks with resources a predicted cost can be determined. This cost is the sum of the values of the time-dependent functions for each task at the time that the resource allocated to it becomes available to perform it. The combination giving the lowest overall cost is then determined.

Additional features are disclosed in the above-mentioned patent which ensure incompatible task/resource combinations are not allocated, and which

to it (for example overnight). However, the result is not readily adaptable to changing circumstances, simply because of the large amount of computer time devoted to preparing it in the first place.

A proposal has been made by G J Garwood and A C Robinson: "Work Management System" in British Telecommunications Engineering Journal: Vol 10 October 1991 to run two different systems, one according to each of the above two approaches: a "real time" system, for dealing with relatively straightforward but urgent cases which are suitable for real time scheduling, and a "schedule building" system for more complex but less urgent cases, which are more suited to a complex but much slower scheduling process. However, this has a number of drawacks. Firstly, an initial decision has to be made as to which resources and tasks to allocate to each system. Resources cannot switch back and forth between systems (for example to perform a short-term task in one system to fill-in between two larger tasks in the other). A link may be provided between the two systems so that if one of the complex schedules fails for an unforseen reason, the resources and tasks which can no longer perform, or be performed in, the original complex schedule are transferred to the real-time scheduling system. However, the real-time scheduling system is not configured to readily deal with such tasks. In particular, the real time scheduler is constrained to only consider tasks whose target time is imminent, as the need to respond quickly precludes inspecting any more tasks.

According to the invention there is provided a task-allocation apparatus comprising:

input means for providing information relating to the tasks to be allocated and the resources available to perform the tasks,

schedule generation means to allocate the resources to the tasks, thereby generating, for each resource, an initial schedule,

storage means for storing the initial schedules,

updating means for receiving, from the input means, updated information relating to the tasks and resources, and

modifying means for modifying the initial schedule of at least a first resource in response to such updated information,

whereby changes to the initial schedules may be made in response to such updated information, independently of the schedule generation means.

arrives. The modification process may be suspended during the periodic generation of the initial schedules, the updated information being used to modify the initial schedules when their generation is complete. Alternatively, the modification process may continue during the generation of the initial schedules, the schedules so modified being input as modifications to the initial schedules when their generation is complete.

If a substantial update data item is received, which would make the existing schedules, or those currently being generated, redundant, the schedule generation process may be initiated at a time other than that determined by the periodicity referred to above.

In a preferred arrangement the schedule generation function comprises a first deterministic stage for scheduling selected tasks, and a second optimisation stage for scheduling the remaining tasks, and wherein the second stage treats the tasks scheduled by the first stage as fixed. In the preferred embodiment the second stage operates according to a stochastic process.

Preferably, groups of linked tasks involving more than one of the resources, or forming a sequence of tasks are selected for scheduling by the first, deterministic, stage.

This architecture allows scheduling to be carried out in several stages, with more changeable, but easier to allocate, tasks being handled in a different manner to tasks which are more difficult to allocate, but less subject to change. The system is conveniently arranged so that optimised schedules are generated periodically, the modification process making short term changes in between the generation of such schedules. This allows the schedule generation process more time to generate each schedule, allowing it to generate a more optimal solution, and/or use more data (e.g. further ahead in time) than would be the case if its run time were constrained by a need to track short term changes in real time.

The architecture described is modular, so that the individual stages of schedule generation and modification can each be adapted or replaced independently of the others.

The terms "deterministic" and "stochastic" in this specification are to be understood as distinguishing between the different methods of operation of the two stages. The deterministic stage operates according to allocation rules which

if the initial optimised schedule of the resource does not contain such a task, determine if a task of that priority exists which has not been scheduled, and select such a task if present.

In this specification the term "scheduled task" means the task currently provisionally allocated to a resource - this task may eventually be allocated to another resource if the schedule is revised. As already discussed, prior art systems allocate tasks to specific resources in advance, to the exclusion of any other suitable resources which, in the event, become available first and have another feasible task allocated. In the system of the present invention, although a high priority task may be scheduled for one specified technician initially, another technician who is suitably positioned and skilled to perform the task may be allocated that task if he calls in first, if to do so produces a net benefit to the schedules. The original schedule for the second technician is then suspended, and each task in that schedule will become unscheduled until a technician suited to the task calls in. This may be the first technician, if his technical skills and geographical location are suitable, and if he calls in before the task is allocated elsewhere. It could be the second technician (i.e. the one for whom it was originally scheduled) if, when he completes the first task, there is still time to perform his originally scheduled task. However, typically the task will be allocated to a third technician, whose own schedule will be interrupted in turn.

To avoid disruption to schedules propagating uncontrollably, the system may be arranged to limit changes to a selected group of resources and tasks. These may be those resources which have related characteristics, such as similar current geographical locations, and/or estimated time to completion of current task, and/or skills, and/or type of tasks they are currently scheduled to perform. A modification to the schedules limited to the area of the "solution space" (the notional multi-dimensional matrix of resources, tasks, time, location etc.) represented by those rescurces ensures that any changes to the schedules will propagate relatively slowly through the solution space, and will therefore be unlikely to result in a total breakdown of the scheduling. In particular, if certain tasks, identified as difficult to allocate, are not permitted to be disrupted, this ensures that "islands of stability" will exist in the solution space, which will tend to reduce the rate at which such disruptions propagate through the solution space.

Figure 5 is a flowchart illustrating the operation of the pre-scheduler 30 of Figure 3.

Figures 6, 7, 8 and 9 illustrate the variation with time of the cost of allocation of a task with time, for five different situations.

Figure 10 is a flowchart illustrating the operation of the optimising subsystem 31 of Figure 3.

Figure 11 is an illustration of the Simulated Annealing process used by the optimising subsystem 31.

Figures 12, 13 and 14 are flowcharts illustrating the method by which tasks are allocated to technicians by the allocation system of Figure 4, based on the initial optimised schedule generated by the system of Figure 3, and real-time modifications to that schedule.

Figures 15, 16 and 17 illustrate three modes of operation of the system of the invention.

Referring to Figure 1, there is shown a telecommunications system, represented by a block N, which is monitored by a fault-monitoring system FMC. The fault monitoring system FMC detects faults in the network which require attention, and also receives inputs from a human operator e.g. to schedule planned maintenance, to generate task requests J1, J2, J3, J4, J5, J6, J7 to be performed by a field force of technicians T1, T2, T3. The task requests are input to a resource allocation system comprising an apparatus in the form of a computer X for allocating resources to tasks, which can communicate through the telecommunications network N with hand-held terminals H1, H2, H3, as required. As shown in Figure 1, terminal H1 is currently in communication with the computer X through a connection C. Each of the hand-held terminals may be a Husky model FS/2 produced by Husky Computers Ltd of Coventry, England, but other suitable equipment may be used.

In a real situation there will be many technicians (typically a few hundred) and tasks. Typically a workforce of one hundred technicians might perform six hundred tasks in one day. Therefore in a typical day approximately 600 tasks will be added to the system, and 600 tasks removed as they are completed. All the new tasks, and a proportion of the completions, will require changes to the day's program. Thus, although each individual technician's schedule only changes a few

- whether the technician can perform each individual task;
- the time the technician would take to travel to the location of each task;
- the time the technician would take to perform each task.
- the relevant importance of each task, determined for example by the number of customers affected and any financial penalties which will be incurred if the task is not performed within a specified time period, or at all; and
- the availability of the other technicians T2, T3.

The availability of these other technicians depends on the times when they each will become available, which in turn depend on the lengths of their current tasks, and the times the technicians started doing them, as well as any travelling time necessary to reach the location of the task from their present locations.

The time that a task will take is subject to some uncertainty, since in many cases tasks involve the investigation and rectification of a reported problem. Until the problem is investigated the time it will take to rectify can only be estimated with a fairly large margin of error. There are also other variable factors, such as the local circumstances of each task, which makes a precise measure difficult. The method used by the program of computer X initially calculates a provisional schedule for each technician, but allows changes to these schedules if a technician reports task completion early, or fails to report at the estimated time, or if new tasks are requested after the provisional schedule has been created.

The method calculates a time-dependent "cost function" for each task. This takes into account the penalty for failing to meet an agreed time. The penalty may be a real monetary cost if compensation is payable to a customer for failures to meet a time, or a 'virtual' cost; e.g. damage to a company's reputation. The penalty is a time-dependent property. In the simplest case the function is zero if the agreed time is met, and a fixed value otherwise. In more complex cases, for example where compensation is payable according to the degree of lateness, it may be some more complex time-dependent function. Scheduling a task earlier than its target time has a contingency value; i.e. if the technician is delayed *en route*, or takes longer on the task than expected, he may nevertheless complete the task before the target time, so a lower cost is appropriate if the task is not scheduled very close to its target time.

customer for a missed appointment, non-productive time can be costed such that the time one is prepared to use to avoid paying that compensation is costed at an equivalent value.

The method determines the combination of the technicians and tasks for which the total of the "technician/task cost" values is a minimum. The cost of *not* allocating a task must also be considered, and this is done by including a non-existent, or "dummy" technician. Other things being equal, if there are more tasks than resources to perform them, the lowest priority task would be scheduled for the dummy technician. For example technician T1 may be currently scheduled to next perform task J5, technician T2 task J7 and technician T3 task J6. Task J4 could be scheduled as a further task for one of these technicians if there is sufficient time remaining in his schedule after the projected completion time of the tasks he has already been scheduled to perform; or otherwise it is scheduled to the dummy technician. When technician T1 reports for instructions, the computer X assesses the current schedule and allocates a task to the technician T1, instructing him over the communications link C. Normally, the task allocated will be the scheduled task (in this case J5), but if a new task (not shown), of higher priority than task J5 is requested, or if the technician T1 reports completion of task T1 unexpectedly early or fails to complete a task at the predicted time, the technician may be allocated a different task, possibly one of the tasks J4, J6 and J7, to ensure the highest priority tasks are still done in time. Technicians T2 and T3 are not given any instructions at this stage as they have not yet completed their current tasks. The scheduling of tasks J6 and J7 to technicians T2 and T3 are provisional, and may be changed, in particular if one of those tasks is allocated to technician T1, or if a high priority task is introduced at short notice.

Various cost analysis algorithms are known for allocating tasks to resources, such as the so called "Hungarian Algorithm" described in a 1955 paper by H W Kuhn "The Hungarian Method for the Assignment Problem" (Naval Research Logistics Quarterly, Vol. 2, pages 83-97) and developed further by M B Wright "Speeding up the Hungarian Algorithm", Computer Operations Research Vol. 17 No 1 pages 95-96 (1990). However the use of these algorithms in real situations is not easy, in particular for inter-related tasks. Moreover, these

some idle time, since the tasks scheduled by the pre-scheduler 30 are only a subset of all the tasks available. In addition the pre-scheduler 30 positions the "next available" time (normally the time that the technician is due to come on duty) breaks, scheduled absences, and the "end of day" event (the time that the technician is scheduled to go off duty) in each technician's tour.

On completion, the results produced by the pre-scheduler are passed to the optimising subsystem 31. The optimising subsystem 31 receives a partial schedule from the pre-scheduler 30 and the data regarding the resources to be allocated, and the tasks not already allocated by the pre-scheduler 30, from inputs 33 and 34 respectively, and generates an initial optimised schedule for passing to a store 32. Both the pre-scheduler 30 and the optimising subsystem 31 follow certain rules, as will be described. The rules may be selected or modified by an operator through respective inputs 35 and 36. the operator also controls the inputs to the pre-processors 33 ,34, to update details of the technicians and tasks.

As will be described, the optimising subsystem 31 generates a provisional schedule of allocations, by initially positioning further tasks around and between the fixed events (including the difficult-to-schedule tasks) established by the pre-scheduler, and then using a stochastic process to re-allocate these further tasks between the different technicians until an optimum schedule is achieved.

The operation of the schedule generation system can be enhanced by periodically halting the operation of the optimisation stage 31, and running a post-optimisation stage 39. This post-optimisation stage uses certain rule-based criteria to assess the schedule developed thus far, identifying those parts which identify which appear close to optimal, adding these to the fixed partial schedule generated by the pre-scheduling stage 30, and then re-running the optimising sub-system 31 again. This directs the optimising subsystem 31 to concentrate on improving those parts of the schedule identified by the post-optimiser 39 as least optimal. This step can be repeated several times.

The provisional schedule finally produced by the schedule generation system 30,31 is then used to programme the real-time modifier 40 illustrated schematically in Figure 4, which is programmed to allocate tasks to technicians according to the provisional schedule, but is capable of departing from the

In the latter case the pre-scheduler 30 will calculate the time the technician is next available, using information on the expected duration of the current activity, plus an estimate of the travelling time that may be involved. In the start of day case, all technicians will be assumed to start work at their scheduled time for reporting on duty.

The resource data includes details necessary to schedule breaks, for example for meals. These breaks are normally of specified duration, but their start (and therefore finish) times are flexible within a specified window to suit the requirements of the work, and their locations are not pre-arranged. Details are stored, for each technician, of the earliest that a break may begin, the latest time it may begin, and its duration. The pre-scheduler 30 will, initially, position each break at its earliest possible start time.

The resource data also includes details necessary to schedule absences from duty. These may be to carry out other duties not controlled by the scheduler, such as training, team meetings etc., as well as authorised non-work absences such as medical appointments. These absences normally have fixed start and finish times, and may also have specified start and finish locations. (These locations may be different if the absence involves travel, for example to take equipment for repair, or to collect supplies). Details are stored, for each technician, of the times and locations of any such scheduled absences. Note that because an absence may have a location associated with it, travel to or from it has to be taken into account in scheduling.

The pre-scheduler is also supplied with details of each technician's end of day data (including any scheduled overtime). The pre-processing will place the end of day or "go home" point so that the technician's day ends at the correct time. The basic fixed points in each technician's schedule have now been created.

At any one time there will be a small portion of the total tasks available for allocation that will be more difficult to schedule, and to move within or between the tours, than the majority of tasks. Typically these tasks will be more heavily constrained than the others. The pre-scheduler schedules these tasks, so that there is the maximum amount of flexibility available for the subsequent scheduling steps. A task scheduled by the pre-scheduler will not be moved to another

5. duration (longest first)

6. number of technicians able to perform the task (smallest number first)

Thus at any time priority will be given to tasks that can be done immediately and, within this category, to the tasks with the greatest importance score.

For each task to be scheduled, the list of technicians who can do the task will be stored into a priority order (step 52). This priority order takes into account factors such as

a. whether the technician has the skills required for each task,

b. whether the technician has any security clearances required to gain access,

c. whether the task is in the technician's preferred working area,

d. success/fail (a measure of whether a task, if allocated to the technician as the next task in his tour after all the other tasks currently in his tour, would still meet his primary commitment target. A success is a task that meets its primary commitment; a fail is one that does not).

e. Success margin (a measure of expected arrival time minus target arrival time (for appointments requiring a response before a predetermined deadline), or estimated completion time minus target completion time (for other tasks). Minimising this margin ensures that lower priority tasks yet to be allocated are not excluded because of higher priority tasks being performed earlier than necessary.

f. The number of skills that the technician possesses (lowest first: to ensure that the most versatile technicians are not allocated to a task which a less versatile technician could have done).

g. Travel time (the time that would be incurred by the technician in getting to the task. If the technician already has tasks or activities in his tour then the travel will be from the latest position to the task. If the task is the first of the day then the travel will be measured from the technician's starting location).

The pre-scheduler 30 then attempts to schedule the tasks to the technicians. Firstly it tries to add the first task to the end of each technician's partial tour in turn (step 53) working through the ordered list of technicians starting with the first (step 54). If the position is valid (step 55) then the task is scheduled to that technician (56). The positioning in the technician's tour takes account of

permitted variations in working hours. If a task may be completed within an individual's permitted overtime then it may be scheduled by the pre-scheduler. However if a task would overrun an individual's overtime limit then it is only scheduled if the task can be split, with the proportion of the task that can be completed before the end of the overtime day being greater than a predetermined minimum. In these circumstances the first part of the task is scheduled to be completed at the scheduled end of day.

The end of day marker for each technician, which has been positioned by the pre-processing, is moved if a task is scheduled that will incur unscheduled overtime. The new position for the end of day marker will be the later of the time the technician completes the task which includes unscheduled overtime, and the time the technician would report off-duty.

It is possible that absences would involve the technician in travel (e.g. travel to and from a team meeting). Such travel is taken into account when determining the technician's expected arrival time for a task, and his expected completion of a task.

The pre-scheduler 30 described above is only used for scheduling the most difficult-to-place tasks. If the pre-scheduling function were used to schedule the entire work programme the run time required would make the schedule unusable by the time it was produced, because of new inputs made to the system during the run time. It is not efficient to devote excessive computer processing time to produce a perfect solution when that solution is likely to be modified as a result of real-time circumstances. For tasks which are easier to schedule, there are likely to be many acceptable, albeit non-optimal, solutions, and it is preferable to obtain a near-optimal solution in a limited time, rather than to produce the perfect solution in a very much longer time. A number of stochastic techniques are known in the art for generating near-optimal solutions to multi-dimensional problems such as the one specified here. Several of these are discussed in the article "Stochastic Techniques for Resource Management" by Brind, Muller & Prosser in the BT Technology Journal Volume 13 No. 1 (January 1995). In particular, this article describes the techniques known as "Hill Climbing", "Simulated Annealing", "Tabu Search" and "Genetic Algorithms".

The optimisation subsystem 31 of the present embodiment will now be described. As shown in Figure 3, the optimising subsystem 31 has three inputs. Firstly, there is an input for a set of tours for the technicians that are available, produced by the deterministic pre-scheduler 30. (In an alternative arrangement, the deterministic pre-scheduler 30 may be omitted and the tours include only fixed points set by the pre-processor 33). This input may also be used for tours generated by the post-optimiser 39 in an iterative arrangement. Secondly, there is an input for the details of the available technicians, 37. Thirdly, there is an input for the details of the unscheduled tasks 38 (i.e. those not selected by the pre-processor 34 for scheduling by the pre-processor 30).

Before starting work the optimising subsystem 31 carries out some pre-processing of the data. This involves working out the earliest and latest that a task may be started. This information is utilised by the optimising subsystem when attempting to add to tours. In addition the pre-processing fixes the activities, breaks, absences and the end of day event in each tour. The optimising subsystem requires various parameter values, programmed by an input 36.

The function of the optimising subsystem 31 is to produce a set of tours for the technicians which minimises the objective cost function. The final tours are produced by making one change at a time to the current schedule, using a single modifying operator. The optimising subsystem passes the details of the tours produced to a store 32, from where they can be retrieved by the real-time modifying system 40.

Note that none of the tasks scheduled by the pre-scheduler 30 can be moved by the optimising subsystem 31 to another technician or to the unscheduled state (dummy technician). However the optimising subsystem 31 will be able to move these tasks within their time windows and insert tasks before, between and after them. It is possible that the final tour produced by the optimising subsystem has tasks, originally positioned by the pre-scheduler, in an amended order (e.g. if the pre-scheduler 30 orders two tasks so that task A is followed by task B, it is possible that the optimising subsystem 31 may insert other tasks between them, which may result in retiming of one or both of the tasks, provided that the various constraints on both tasks are still complied with, and their order is preserved.

Taken together these three costs have the effect of ensuring that the simulated annealer tries to minimise travel, allocations which do not reflect the required skill bias, and overtime.

The cost of allocation is a function of the type of task, importance cost of the task, and whether the task is positioned so that it meets a defined deadline. In general this takes the form of reducing the cost of allocation the earlier the task is completed. This is calculated, for tasks where there is a target arrival time, as the difference between the expected arrival time and the target arrival time, and for tasks where there is a target completion time as the difference between the estimated completion time and the target completion time.

These terms are modified to give the function two important further properties as follows. Firstly, a property "P" is defined as the ratio of the difference between the expected time of meeting the target and the target itself, and the maximum time that the expected time may exceed the target. For example a task with an appointment to be made in the period from 10.30am to 1pm, where the expected arrival time is 11.30am has a value of P which is 11.30am minus 1pm (90 minutes) divided by 10.30am minus 1pm (i.e. 150 minutes) equals 90/150=0.6, as does a task that has to be completed by 5pm and which is expected, at 12 noon, to be actually completed by 2pm. In both cases the target is met 40% of the way into the available window and the only difference is the cost of allocating, dependent only on the importance cost of allocating each task.

Secondly, the cost of delaying a task positioned near to the point at which it is going to fail should be greater than the cost of bringing forward, by the same amount of time, a similar task which is still a long way from the time at which it will fail. For example if there are two tasks with a commitment time of 5pm, where one was expected to be completed at 4.50p.m. and the other at 12.00 noon, then a move which resulted in the first being brought forward by five minutes while the second is delayed by five minutes reduces the total scheduling cost. Thus moves that delay a task already close to failing will only improve the objective function if the delay allows a very much larger benefit elsewhere.

The only difference between the cost of allocating the different tasks at the start of their respective windows is due to any differences there may be in the

Values of IMU and IEX greater than 1 increase the differential cost of allocating tasks with a high importance score compared to a low score.

Infill tasks are costed in the objective function, using a cost of zero for each task scheduled and a cost of 1 for each task unscheduled. This rule is designed to ensure that it will always be cheaper to schedule an infill task rather than to leave it unscheduled, but it will never be worth delaying a more important task to enable an infill task to be scheduled.

The objective function uses the following parameters:

- ETT (Estimated travel time in minutes: generally the time estimated for the technician to travel from one task to the next).

- FTF (Failed Task Flag: = 1 if task fails its commitment target, 0 otherwise)

- FSP (Failed secondary commitment penalty)

- SCT (Secondary commitment time: a time, later than the target completion time, at which the cost penalty has a step change)

- TSS (Time at which the run of the search system starts.)

- ETA (Estimated time of arrival: calculated as the time the technician will arrive on site assuming that all travel and task durations are exact and that tasks are scheduled as soon as the previous task finishes, if this does not move the task out of its time window, or at the time such that the technician arrives at the exact start of the task's time window if that is later. A task's time window is defined as the time between the task's earliest start time (EST) and latest start time (LST).)

- TAT (Target arrival time: given by the task pre-processor 34. For an appointed task this will be the end of the appointment slot)

- UOT (Amount of unscheduled overtime in minutes: the amount of time beyond the technician's scheduled end of day required to perform the schedule).

- OTP (Unscheduled overtime penalty: this parameter takes a value greater than or equal to zero, and is used to work out the contribution to the objective function that each task allocation incurring unscheduled overtime will make. The default value is zero).

- UAP (Unallocated appointment penalty: this parameter takes a value greater than or equal to zero, and is used in working out the contribution to the objective function for appointments which are not allocated. The default is again zero).

- FTP (Failed task penalty: takes a value greater than or equal to zero, and represents the amount that will be used in working out the contribution to the objective function for non-appointment tasks. Default is zero)

- SBP (Skill bias penalty: takes a value greater than or equal to zero, and represents the amount to be added to the objective function for each task allocation that does not reflect the desired skill bias; i.e. if the skill bias flag (SBF) is set to 1. The default value of the parameter is zero).

- ATC (Arrival type constant: an integer greater than zero: this represents the period of time over which the cost of allocating a failed "arrival-type" task - i.e. one where the commitment is to arrive at site at a given time, as distinct from one where the commitment is to complete the task - doubles)

- ITP (Infill task travel penalty: takes a value greater than or equal to zero and represents an amount which will be used to work out the contribution to the objective function for the travel associated with each infill task allocated. The travel contribution is calculated as ITP multiplied by ETT (estimated travel time). The default value of the parameter is zero.)

- MIT (Maximum infill travel: takes a value greater than or equal to zero, and represents the amount of travel beyond which the cost of allocating an infill task exactly equals the cost of not allocating it. The higher the value, the further a technician might be allowed to travel to such an infill task.)

Equation 2: For allocated arrival type tasks where ETA minus TAT is less than or equal to zero, the contribution to the objective function is:

$(1 + P/2) \times P \times (IMU^{IEX} \times IMP) + (ETP \times ETT) + (OTP \times UOT) + (SBP \times SBF)$

where $P = (ETA - TAT)/(TAT - TSS)$ [if $TAT = TSS$ then $P = 0$]

Equation 3: For allocated arrival type tasks, where ETA - TAT is greater than zero, the contribution to the objective function is:

$(P + FTP) \times (IMU^{IEX} \times IMP) + (ETP \times ETT) + (OTP \times UOT) + (SBP \times SBF)$

where $P = (ETA - TAT)/ ATC$

Equation 4: For deallocated appointments and arrival-type tasks the contribution is:

$(IMU^{IEX} \times IMP) \times UAP$

Equation 5: For allocated commitment tasks where ETC - TCT is less than or equal to zero the contribution is:

$(1 + P/2) \times P \times (IMU^{IEX} \times IMP) + (ETP \times ETT) + (OTP \times UOT) + (SBP \times SBF) + (BTP \times BTF)$

where $P = (ETC - TCT) / (TCT - TSS)$

Equation 6: For allocation commitment tasks where ETC - TCT is greater than zero, but the secondary commitment limit is not exceeded, the contribution is:

$(P + FTP) \times (IMU^{IEX} \times IMP) + (ETP \times ETT) + (OTP \times UOT) + (SBP \times SBF) + (BTP \times BTF)$

where $P = (ETC - TCT) / (SCT - TCT)$

Equation 7: For allocated commitment tasks where ETC - TCT is greater than zero and the secondary commitment limit is exceeded, the contribution is:

$(P + FSP + FTP) \times (IMU^{IEX} \times IMP) + (ETP \times ETT) + (OTP \times UOT) + (SBP \times SBF) + (BTP \times BTF)$

where $P = (ETC - TCT) / (SCT - TCT)$

Equation 8: For unallocated commitment tasks the contribution is to be:

$\{(ETC - TCT) / (SCT - TCT) + (FTP \times FTF) + (FSP \times FSF)\} \times (IMU^{IEX} \times IMP)$

twice the importance score (i.e. 100) beyond a secondary commitment time (in this case the penalty time; 2040 minutes i.e. 10 a.m. on Day 2).

Figure 9 shows how the cost of allocation varies with time for a task with a target arrival time with an importance score of 900. In this figure the target arrival time is 12 noon (720 minutes) and the value of the parameter FTP is 0.1. This results in a step change of 90 (i.e. 0.1 times the importance score of 900) when the task passes 12 noon.

An initial schedule is built up by taking the partial schedule generated by the pre-scheduler 30 and arbitrarily allocating further tasks to technicians. This initial schedule is then modified by the optimising subsystem 31. This process is illustrated in the flow chart of Figure 10. The process includes four steps 1001, 1002, 1005, 1016 which require the generation of a random number. The tasks and technicians are each allocated a number.

The process starts by costing the original schedule (step 1000). Next, a random number generator is used to select one of the tasks. With the exception of the pre-allocated tasks, already discussed, each task has the same probability of being selected, whether it is currently in a schedule or not. How the process continues, once a task has been selected, depends on whether the task selected is already in a tour.

A feasible technician (i.e. one who can do the task as determined by the pre-processor 33) is also selected at random (using an analogous process to that used for selecting a task). The number is selected (step 1002) from the range 1 to N + 1, where N is the number of feasible technicians: however the number which corresponds to the technician to whom the selected task is currently scheduled is excluded from selection. The number N + 1 represents a "dummy" technician. Allocating a task to the dummy technician constitutes deallocating the task for the purposes of the objective function. Note that the chance of scheduling a task to the dummy technician is 1/N, and therefore diminishes as the number of technicians in the system increases. If the task is not already scheduled, then it is the dummy technician who is excluded from selection.

If the task is not for the dummy technician (step 1003), and not for an appointed time (step 1004), a random position in the tour of the selected technician is selected (again using the random number process; step 1005) and the

previous best, the solution is saved as the new "best value" (step 1020) for future comparisons. Any change may be rejected (step 1018) either at this final stage, or because the task cannot be inserted into the schedule (step 1007 or 1009).

The process is then repeated for another task (step 1001) using either the revised schedule if the change was accepted (step 1015), or the previous schedule if the change was rejected (step 1018). Note that a move which is accepted in step 1015 is used as the basis for the next iteration, whether or not it is an improvement on the "best value" (steps 1019/1020).

Deallocating a task (i.e. scheduling it to the dummy technician) will always increase the objective function, but such deallocations are accepted with probability p, thereby allowing the possibility, in the next iteration of the process, to reschedule a replacement task to the technician from whom the task was taken. Note that there are no skill, time, or other constraints on allocating a task to the dummy technician - such an allocation is always feasible, but always increases the objective function.

The probability p of accepting a move that makes the value of the objective function increase is given by the formula $p = e^{(-\mathrm{delta/temperature})}$

The "temperature" is a concept which controls the number of moves that are made that increase the value of the objective function. The higher the value of temperature the more moves will be accepted that increase the value of the objective function. During the search the value of temperature is gradually reduced, so that at later points of the search fewer such moves are accepted. Delta is the difference between the value of the objective function after the change and the value before the change. After a given number of moves at a given temperature the value is reduced. The reduction is produced by multiplying the temperature by a predetermined value. All moves, whether they be valid or invalid, rejected or accepted, are included in this count.

The relationship between delta, temperature and the probability of accepting a move that makes the value of the objective function worse (i.e. greater) is illustrated in Figure 11. This figure shows the two key features of the cooling regime; firstly that the probability of accepting a move that makes the value of the objective function worse decreases as the magnitude of delta increases, and secondly that as the temperature decreases so does the probability.

identify individual schedules which appear to be close to optimal, such as schedules having only small amounts of idle time or travel time, and which do not involve an itinerary requiring doubling-back. Such schedules are identified as fixed, and the optimisation process resumed. This ensures that the optimisation process concentrates on those parts of the overall schedule in which improvements are most likely to be found, by constraining its search to those areas of the "solution space".

After a suitable run-time, a final set of technician schedules is produced. This is then passed to the real-time schedule modifier 40. Whilst the real-time modifier is running, using this schedule, the schedule generator 30,31 can resume operation in order to generate a new schedule using updated data.

The operation of the real-time modifier 40 (Figure 4) is shown in flow chart form in Figures 12, 13 and 14. Figure 4 shows in block diagram form the principal features of the real-time modifier 40. A schedule status register 42 stores the current status of the schedules, which are initially supplied, via the schedule store 32, from the optimising subsystem 31 (see Figure 3). A technician status register 43 and pool of work register 44 similarly store data relating to the technicians and tasks respectively, initially obtained from the respective pre-processors 33, 34. These three registers are all updatable, as will be explained. A parameter input 41 allows an operator to set the various weightings and other values used by the system.

The technician status register 43 is updatable by inputs from the technicians T1, etc themselves, in particular to record a technician's status as on-line or off-line. The pool of work register is also updatable by means of a manual interface (a computer terminal) 45, which allows an operator to add new tasks to the pool of work during a run of the real-time allocator, e.g. in response to a customer reporting a fault. Automatic inputs, connected to fault monitors within the network 46, may also be provided.

The registers 42, 43, 44 all provide inputs to an allocation processor 47, which uses the current status information in the registers in a manner to be described with reference to Figures 12, 13 and 14 to generate an allocation for a technician T when he comes on-line to request instructions. The resulting allocation is passed to an instruction generation unit 48, which transmits

"Pending": the scheduling status of an activity (e.g. task, absence, break) which is scheduled for a technician but not yet allocated to him, and for which the information on which the schedule was based is assumed to be still accurate (i.e. has not been marked "inaccurate").

The processes illustrated in Figure 12 to 14 will now be described. In outline these are as follows.

Figure 12 shows the process for determining what instruction to give a technician who has just checked in (step 1200). The instruction (1211 to 1215) will usually be to carry out a task (1213), but other instructions such as taking an absence (e.g. to attend a meeting or training session, 1211), may also be issued.

Figure 13 is the process by which a suitable task is selected.

In both the processes of Figures 12 and 13, a subroutine is used which determines the feasibility of the technician performing a given task. This process is illustrated in Figure 14.

Figure 12 illustrates the process for determining the allocation of a task for an on-line technician. When a technician checks in (step 1200) the system consults the schedule to identify the next activity for the technician (1201). If the next scheduled activity is "end of day" (i.e. the technician has no further activities already scheduled for the rest of the day), the system steps to process 13 to try to find a suitable task for the technician. If there are scheduled activities before the "end of day" activity (negative outcome to step 1201) a "preselection absence check" process is performed (steps 1202 to 1207). Firstly, the system checks the technician's schedule to determine if any absences are scheduled for the technician (step 1202). If there are no such absences then a next task for the technician is found using the process 13 to be described below.

If there is at least one absence then the system checks to see if the next absence is due to start within a predetermined time after the current time (step 1203). If this is the case then a check is made to establish whether the scheduled end of the absence will fall within a period within which a break must be taken (step 1204). If it will not, the technician is instructed to take the absence, e.g. to attend a meeting (step 1211), and to report back for further instructions afterwards. The database is updated (step 1216) by recording the start of the period of absence, the technician's next expected contact time is updated to be

for him to perform, and takes into account in such an optimisation even tasks which have been added to the system since the last schedule revision. This allows more time between rebuilds of the entire schedule, and therefore allows more time to be spent on each rebuilding run.

Allocation of a task to a technician other than the one scheduled to do it will only modify the schedule with respect to that technician and, if the task was originally scheduled for another technician, that other technician. Periodically during the day, a new schedule is built up. The frequency with which this is done should be selected according to the rate at which changes take place, such that the previous schedule is not totally disrupted by on-line changes before the new one is available. However, the more time that can be allowed for the generation of each new schedule, the more optimal the solutions which can be generated.

If possible, the initial schedule used in the first iteration of the optimisation stage 31 may be the current schedule, generated by the real-time optimiser 40 on the previous cycle, subject to any new fixed points added by the deterministic stage 30.

The details of the tasks which have not yet been allocated, (whether provisionally scheduled for allocation or not) are stored in the "pool of work" database 44. When considering a technician for allocation to a task the following factors are determined, to as to be used in the various repair and optimisation checks that take place:

a. The location from which the technician is deemed to be travelling to attend the task under consideration, which is considered to be the location at which the technician's current task is, or his start of day location if no task has yet been allocated.

b. Allocation start time (i.e. current time or the start of the working hours of the technician, whichever is the later).

c. The time remaining of the technician's working hours, including any scheduled overtime.

d. The location at which the technician must end his working period, which may be his home location or the location of a scheduled absence.

e. A preferred working location, and a travel radius surrounding it. Tasks located outside this radius will not be allocated to the technician.

allocate. If so, it is allocated to the technician without any further optimisation being attempted (outcome 1308).

In all other cases (i.e. if there is no feasible task in the schedule with such a flag), the process investigates (steps 1304/1304A) whether there is a task in the pool of work 44, flagged as very important and near to its commitment time, which has not yet been scheduled. This may occur for several reasons; for example because it is a new task, or one which has been lost from another schedule elsewhere.

For reallocation of a task to the currently on-line technician the four conditions described with reference to step 1302 must all be satisfied, and also the following:

e. the task is not "firmly scheduled" to the scheduled technician (if any) by the pre-scheduler 30 ;

f. the technician under consideration has the necessary skills and permissions to do the task;

g. the location of the task is within the technician's geographical limits.

h. the technician has the appropriate skills and permissions;

i. the task is within a predetermined period of its target time,

j. the task can be performed before the technician's next fixed task.

All tasks so selected are then ordered in the following sequence;

1. Any task specifically pre-scheduled to the on-line technician.
2. Travel time to task (in increasing order).
3. Skill preferences.
4. Priority (in decreasing order)
5. Time to commitment.

Travel time may be computed from one location to another location using any known route navigation system, taking into account factors such as type of roads, time of day, etc. Where a task is scheduled but the previous task is not yet scheduled, a typical travel time factor for tasks in the general area is used. If the task is marked as "inaccurate", or a repair operation is being performed on the schedule, then the travel time must be recalculated, since the estimate for the scheduled entry will no longer be correct.

location within the appointment slot, if any. The system therefore tests firstly whether the task has an appointment timeslot (step 1402) and if so whether the technician can arrive in that slot (i.e. neither too early, which would incur idle time, nor too late: steps 1403 and 1404). If he can arrive within the appointment timeslot, or if there is no appointment timeslot, the task is feasible (1410). If he cannot arrive within the appointment slot, the task is infeasible (1413).

If the technician has a break yet to be taken, there are four possible outcomes of the test (shown as 1410 to 1413 in Figure 14). Either the task is infeasible, (1413) or it is feasible; in the latter case it may be necessary to schedule a break before (1412) or during (1411) the task, or not at all (1410) if the task can be completed before a break has to be taken.

If the technician has not yet taken a break, then the test in step 1401 returns a positive result. In this case, the next test is again to check whether the task has an appointment time (step 1402a). If it has, then a test is made as to whether the technician is going to be too early for the appointment (step 1403a). If the technician is not going to be too early, a test is made to check whether the technician would arrive after the end of the appointment slot (step 1404a). These three tests are essentially the same as in steps 1402, 1403, and 1404, discussed above, but they lead to different outcomes. If the outcome of test 1404a is positive (in other words the technician cannot arrive until after the end of the appointment slot) the task is not feasible for that technician (outcome 1413) and a different task must be assigned to the technician. If the technician will neither be too early nor too late (outcomes of tests 1403a and 1404a both negative) then a test (step 1405) is made to check whether the task can be completed before the latest time at which the break may be started. If the outcome of this test is positive then the task is feasible, and no instructions for break are required (outcome 1410) . However if the task cannot be completed before the latest time at which a break must be taken (outcome of test 1405 negative) then the next test (step 1406) is made, to determine whether it is still too early for the technician to take his break immediately. If the outcome of this test is positive, then the task is feasible but the break must be taken during the task (outcome 1411), because the outcome of test 1405 was negative, so the task cannot be completed before the end of the period within which the break must be taken.

Returning now to Figure 12, at step 1208, if a suitable task was found by the process of Figure 13 (outcomes 1305, 1307, 1308) this task is allocated to the technician (step 1213), together with any other instructions generated in the feasibility test (Figure 14), e.g. to take a scheduled break before the task (1412) or during the task (step 1411). However if no valid task is found (outcome 1309) then a test is made as to whether a break can be taken (step 1209). If no break is scheduled, the technician has no work to do and is instructed to contact supervisor for instructions (step 1214), either to be authorised to finish work for the day or to await further instructions, for example in the event that a new task may come in, or enter its appointment window.

If a break can be taken (step 1209) then a test is made (step 1210) as to whether, if an absence is scheduled but not yet due to start, that absence can be taken immediately after the break. This is a test as to whether, by the time the break period has finished, the absence will be due to be taken. (Note that if the absence is already due to start the outcome of test 1203 will have been negative, so test 1204 will have been performed instead of test 1210). Depending on the outcome of this test (step 1210) the technician is instructed to take the break only (step 1215), the absence remaining in his schedule to be taken later; or to take the absence and the break together (step 1212). If no absence is scheduled (outcome of test 1202 negative) then the answer to the enquiry in step 1210 is of course always in the negative and so outcome 1212 is not relevant.

Once a task has been allocated, with or without a break, (outcomes 1211 1212, 1213 or 1215) ,or a decision is made that no task or other activity such as an absence or break can be allocated, (step 1214) the technician is instructed accordingly. The instructions are generated by a message generation unit 48. This may generate a display of the allocation for use by a human dispatcher to pass the instruction on to the technician T, or by transmitting a data message directly to the technician's handset (e.g. H1) over the communications link C. The revised schedule details are stored in the databases 42, 43, 44 (step 1216). For the technician register 43 these detail the current location of the technician, and the predicted completion time (i.e. the time at which the technician is next expected to come on-line requesting new instructions.) This will in general be the sum of the

In Figure 15, the schedule modification process 4 is suspended during the initial schedule generation process 3. Inputs 1a occurring whilst the initial generation process process 3 is in operation are not immediately processed according to the current modification process 4, but are buffered in a store 5 until the process 3 is complete, and then processed according to the modification process 4a based on the new initial schedule, providing an output 2a.

Alternatively, in Figure 16, an output 2a is generated immediately on receipt of an input 1a, using the modification process 4 using the existing data, but the input 1a is also buffered to provide an input to the new modification process 4a when the initial schedule 3 is complete. This has the advantage of providing a prompt response 2, whilst the arrangement of Figure 15 has the advantage of using a more recent initial schedule to in generating the response 2a. Selection of one or other arrangement depends on the acceptable delay between an input 1a and an output 2a.

If data 1b is provided which requires substantial modification to the initial schedule, such that solutions 3,4 generated using the existing initial schedule will be invalid, the schedule modification process 4 and the initial schedule generation process 3 are suspended (if currently running) and the initial schedule generation process then restarted (3a) with the new data. Inputs 2a received during running of the restarted process 3a are buffered (5) until the new schedule is generated.

5. Apparatus according to claim 3 or 4, wherein the second stage operates according to a stochastic process

6. Apparatus according to claim 5, wherein the stochastic process is a Simulated Annealing process.

7. Apparatus according to claim 3, 4, 5 or 6, wherein the schedule generation means comprises a third, post-optimisation stage, comprising means for analysing the schedules created by the second stage, means for identifying schedules requiring further optimisation, and means for inputting such schedules into a further iteration of the second stage for further optimisation, the further iteration of the second stage having means to treat the schedules not so identified as fixed.

8. Apparatus according to any preceding claim, wherein the schedule modifying means comprises a plurality of selection means, each for assessing in turn the plurality of tasks waiting to be performed to determine if a task of a given priority suitable for performance by the first resource is available, and allocating such a task to the first resource, the selection means being arranged to identify tasks of successively descending priority, such that tasks of high priority are allocated in preference to lower priority tasks in the initial optimised schedule for the first resource.

9. Apparatus according to claim 8, wherein at least one of the selection means comprises first assessment means for determining if the initial optimised schedule of the first resource includes a task of the given priority, and selecting said task if present, and second assessment means, operable if the initial optimised schedule of the first resource does not contain such a task, for determining if a task exists which has not been scheduled, and selecting said task if present.

10. Apparatus according to any preceding claim, wherein the schedule modifying means comprises means to identify resources having characteristics related to those of the first resource, and arranged to modify the schedules of only

16. Method according to claim 12, 13, 14, or 15 wherein the schedule generation process is initiated if a substantial update data item is received.

17. Method according to Claim 12, 13, 14, 15, or 16 wherein the schedule generation function comprises a first deterministic stage for scheduling selected tasks, and a second optimisation stage for scheduling the remaining tasks, wherein the second stage treats the tasks scheduled by the first stage as fixed.

18. Method according to claim 17 in which groups of linked tasks involving more than one of the resources, or forming a sequence of tasks are selected for scheduling by the first, deterministic, stage.

19. Method according to claim 17 or 18, wherein the second stage operates according to a stochastic process

20. Method according to claim 19, wherein the stochastic process is a Simulated Annealing process.

21. Method according to claim 17, 18, 19 or 20, wherein the schedule generation function comprises a third, post-optimisation stage, in which the schedules created by the second stage are analysed to identify schedules requiring further optimisation, and such schedules are input into a further iteration of the second stage for further optimisation, the further iteration of the second stage treating as fixed the schedules not so identified.

22. Method according to any of claims 12 to 21, wherein the schedule modifying process comprises a plurality of selection steps, in each of which the plurality of tasks waiting to be performed is assessed to determine if a task of a given priority suitable for performance by the first resource is available, and such a task is allocated to the first resource if identified, the selection steps each being arranged to identify tasks of successively descending priority, such that a task of

Fig.1.

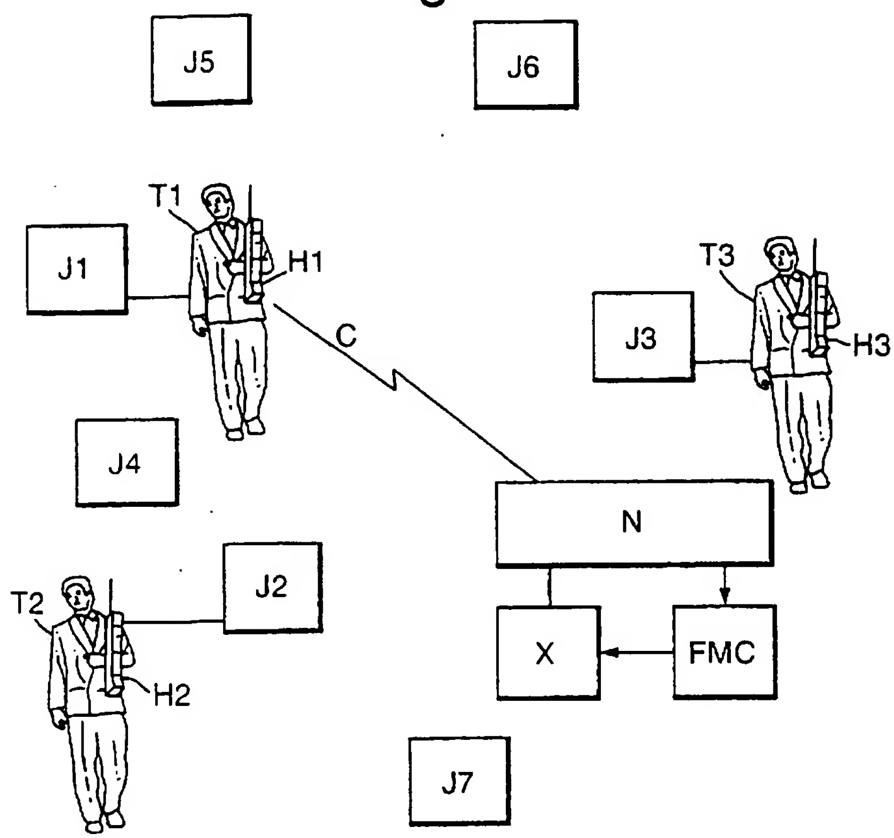

J5
J6
T1
J1
H1
C
T3
J3
H3
J4
N
T2
J2
X
FMC
H2
J7

Fig.2.

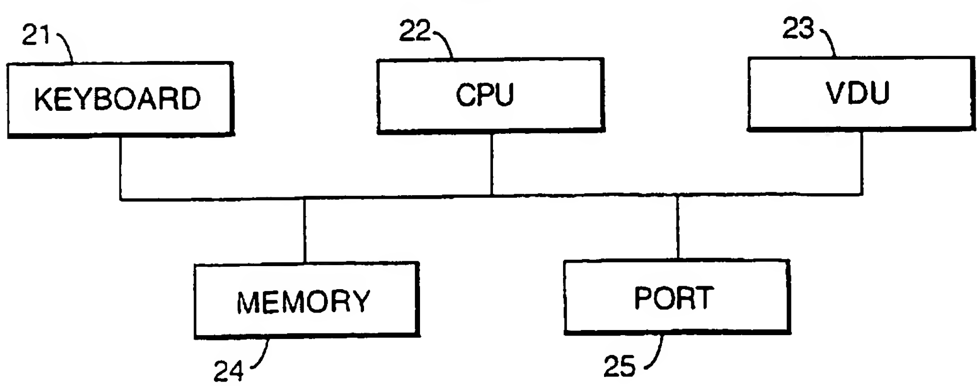

21
KEYBOARD
22
CPU
23
VDU
MEMORY
24
PORT
25

Fig.4.

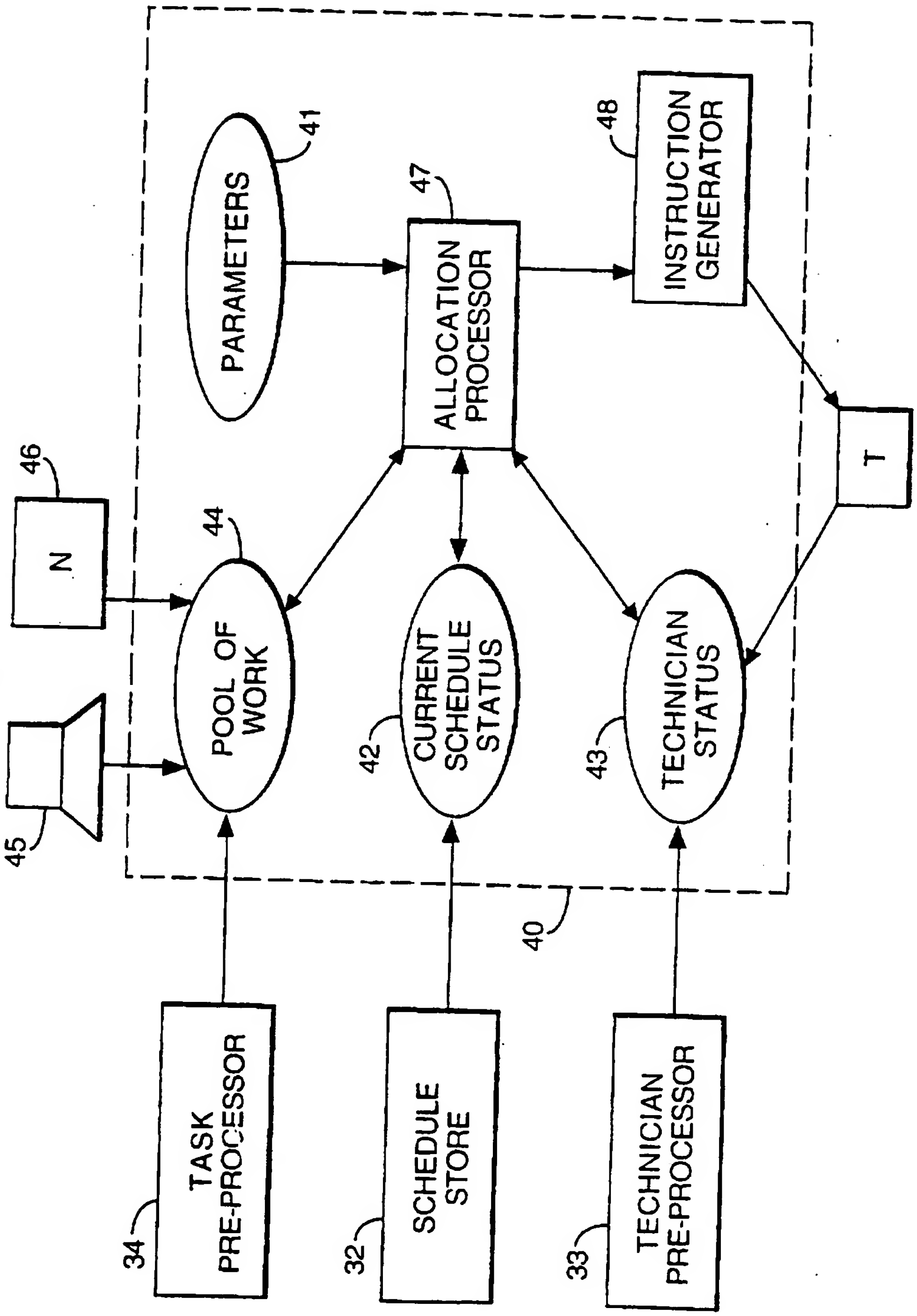
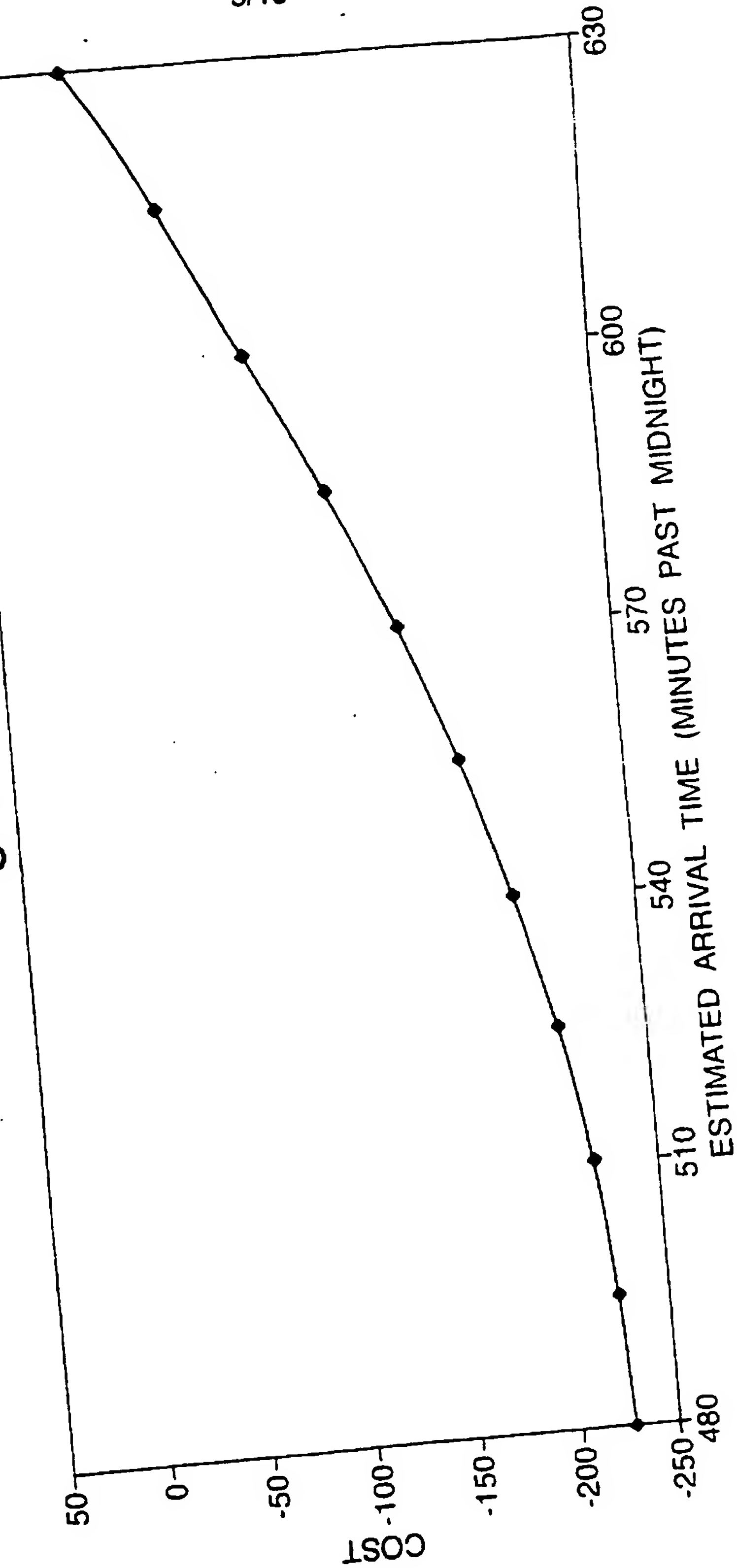

PARAMETERS
41
ALLOCATION PROCESSOR
47
INSTRUCTION GENERATOR
48
N
46
POOL OF WORK
44
CURRENT SCHEDULE STATUS
42
TECHNICIAN STATUS
43
T
45
40
TASK PRE-PROCESSOR
34
SCHEDULE STORE
32
TECHNICIAN PRE-PROCESSOR
33

Fig.6.


COST
50
0
-50
-100
-150
-200
-250
480
510
540
570
600
630
ESTIMATED ARRIVAL TIME (MINUTES PAST MIDNIGHT)